



---

A Sierra Monitor Company

---

**Driver Manual**  
**(Supplement to the FieldServer Instruction Manual)**

**FS-8700-72 Secutron Modul-R**

**APPLICABILITY & EFFECTIVITY**

**Effective for all systems manufactured after August 2008**

<b>Driver Version:</b>	<b>1.01</b>
<b>Document Revision:</b>	<b>8</b>

## TABLE OF CONTENTS

1.	SECUTRON MODUL-R DRIVER DESCRIPTION .....	3
2.	DRIVER SCOPE OF SUPPLY.....	3
2.1.	Supplied by FieldServer Technologies for this driver.....	3
3.	HARDWARE CONNECTIONS .....	4
4.	CONFIGURING THE FIELDSERVER AS A SECUTRON MODUL-R DRIVER CLIENT ..	5
4.1.	Data Arrays/Descriptors .....	5
4.2.	Client Side Connection Descriptors.....	6
4.3.	Client Side Node Descriptors .....	6
4.4.	Client Side Map Descriptors.....	7
4.4.1.	<i>FieldServer Specific Map Descriptor Parameters</i> .....	7
4.4.2.	<i>Driver Specific Map Descriptor Parameters</i> .....	7
4.4.3.	<i>Timing Parameters</i> .....	7
4.4.4.	<i>Map Descriptor Example</i> .....	8
5.	CONFIGURING THE FIELDSERVER AS A SECUTRON MODUL-R DRIVER SERVER	9
5.1.	Data Arrays .....	9
5.2.	Server Side Connection Descriptors .....	10
5.3.	Server Side Node Descriptors .....	10
5.4.	Server Side Map Descriptors .....	11
5.4.1.	<i>FieldServer Specific Map Descriptor Parameters</i> .....	11
5.4.2.	<i>Driver Specific Map Descriptor Parameters</i> .....	11
5.4.3.	<i>Map Descriptor Example</i> .....	12
	APPENDIX A. ADVANCED TOPICS.....	12
	Appendix A.1. Event Function or Condition Codes.....	12
	Appendix A.2. Panel Circuit or Zone numbers.....	13
	Appendix A.3. Panel Power-up, Restarts and Resets .....	14
	Appendix A.4. Driver Messages, Error Messages and Optional Map Descriptors.....	14

## 1. Secutron Modul-R driver Description

The Secutron Modul-R driver allows the FieldServer to transfer data to and from a Client and a Server panel on a single point to point network over RS-232 using Secutron Modul-R protocol.

The protocol is strictly command/response with the Client polling a Server device. A Server device can report the generation and clearing of alarms to the polling Client. One panel can be connected per FieldServer port. The driver supports networked panels connected to a single port. The server driver only supports 10 panels due to memory limitations. A Server panel can consist of up to 100 circuits.

When an alarm is generated, a bit is set and when the alarm is cleared, the bit is cleared. The panel number is identified by the Node\_ID in the configuration file.

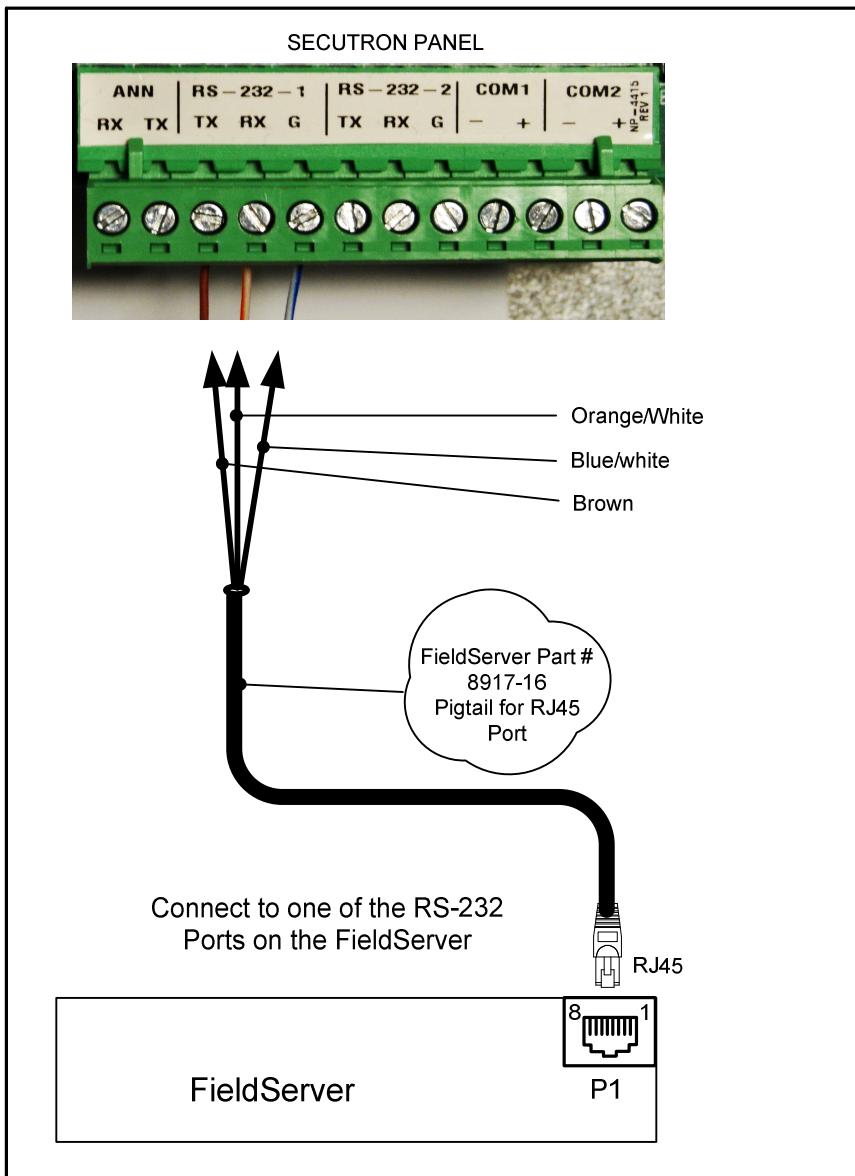
## 2. Driver Scope of Supply

### 2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8917-16	Pigtail cable for RJ45 Port
FS-8700-72	Driver Manual.

### 3. Hardware Connections

The FieldServer is connected to the panels as shown below. Configure the panels according to manufacturer's instructions.



Connector Pinouts

RJ-45 Pin	From RJ-45	To Secutron
1	Brown	Tx
8	White/Orange	Rx
5	Blue/white	GND

#### 4. Configuring the FieldServer as a Secutron Modul-R driver Client

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See ".csv" files on the provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with Secutron Modul-R panels.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Secutron Modul-R driver communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the destination device addresses need to be declared in the "Client Side Nodes" section, and the data required from the Servers needs to be mapped in the "Client Side Map Descriptors" section. Details on how to do this can be found below.

Note that in the tables, \* indicates an optional parameter, with the bold legal value being the default.

##### 4.1. Data Arrays/Descriptors

Section Title	Function	Legal Values
Data_Arrays		
Column Title		
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Format	Provide data format. Each Data Array can only take one format.	Bit, Byte, Packed_Bit, Uint16
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required for the data being placed in this array.	1-10,000

##### Example

```
// Data Arrays
Data_Arrays
Data_Array_Name,
p1c1_devices,
p1c1_function,
p1c69_restart,
p1c69_function,
p1c90_reset,
p1c90_function,
```

	Data_Array_Format,	Data_Array_Length
p1c1_devices,	Bit,	1
p1c1_function,	Byte,	1000
p1c69_restart,	Bit,	1000
p1c69_function,	Byte,	100
p1c90_reset,	Bit,	100
p1c90_function,	Byte,	100

The Poller Data Array will contain non-zero data when communications with the panel is OK.

#### 4.2. Client Side Connection Descriptors

Section Title	Function	Legal Values
Connections		
Column Title		
Port	Specify through which port the device is connected to the FieldServer	P1-P8 <sup>1</sup>
Baud*	Specify baud rate	150, 300, 1200, 2400, 4800, <b>9600</b> (Vendor limitation)
Parity*	Specify parity	<b>None</b> (Vendor limitation)
Data_Bits*	Specify data bits	<b>8</b> (Vendor limitation)
Stop_Bits*	Specify stop bits	<b>1</b> (Vendor limitation)
Protocol	Specify protocol used	Secutron
Handshaking*	Specify hardware handshaking	None

##### Example

// Client Side Connections				
Connections				
Port, P1,	Protocol, Secutron,	Baud, 9600,	Parity, None,	Handshaking, None

#### 4.3. Client Side Node Descriptors

Section Title	Function	Legal Values
Nodes		
Column Title		
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	Generic Node ID unique to port. Also used as panel number.	0-999
Protocol	Specify protocol used	Secutron
Port	Specify which port the device is connected to the FieldServer	P1-P8 <sup>1</sup>

##### Example

// Client Side Nodes				
Nodes				
Node_Name, Node_1,	Node_ID, 1,	Protocol, Secutron,	Connection, P1	

<sup>1</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

#### 4.4. Client Side Map Descriptors

##### 4.4.1. FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in "Data Array" section
Function	Function of Client Map Descriptor	Rdbc, Passive_Client
Node_Name	Name of Node to fetch data from	One of the node names specified in "Client Node Descriptor" above
Length	The number of devices on the circuit the Map Descriptor must detect	0-999
Block_Type*	Used to enable packed bit when using a Packed_Bit Data Array on the Map Descriptor	Flag, -

##### 4.4.2. Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Secutron_Type	Identifies the type of data the Map Descriptor points to	Poller, Device, Function
Secutron_Circuit	Identifies the circuit or panel zone	0-99

##### 4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	>= 0s

#### 4.4.4. Map Descriptor Example.

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name,	Secutron_Type,	Secutron_Circuit,	Length
P1_poller,	0s,	poller	0,	Rdbc,	Node_1,	-	-	1
P1c1_devices,	-	P1c1_devices	0,	Passive_Client,	Node_1,	-	-	200
P1c1_function,	-	P1c1_function	0,	Passive_Client,	Node_1,	-	-	200
P1c69_restart,	-	P1c69_restart,	0,	Passive_Client,	Node_1,	-	-	100
P1c69_function,	-	P1c69_function,	0,	Passive_Client,	Node_1,	-	-	100
P1c90_reset,	-	P1c90_reset,	0,	Passive_Client,	Node_1,	-	-	100
P1c90_function,	-	P1c90_function,	0,	Passive_Client,	Node_1,	-	-	100

A Data Array name must be one found under Data\_Arrays. Data from a circuit event will be stored in the array at Data\_Array\_Offset.

Any unique name. Name will appear in Map Bridge Map Descriptor status information screens.

Function must be read for pollers and all others Passive\_Client.

The type of data the Map Descriptor points to. Device data contains alarm statuses. Function data contains condition codes. Refer to Appendix A.1 for more information.

The circuit or zone number used to trap incoming events into the Map Descriptor. See Driver notes.

The number of devices on this circuit this Map Descriptor can detect for incoming events.

The rate at which the driver polls the panel.

## 5. Configuring the FieldServer as a Secutron Modul-R driver Server

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See ".csv" files provided with the FieldServer).

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Secutron Modul-R driver communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the FieldServer virtual node(s) need to be declared in the "Server Side Nodes" section, and the data to be provided to the Client needs to be mapped in the "Server Side Map Descriptors" section. Details on how to do this can be found below.

Note that in the tables, \* indicates an optional parameter, with the bold legal value being the default.

### 5.1. Data Arrays

Section Title		
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Format	Provide data format. Each Data Array can only take on one format.	Bit, Packed_Bit, Byte, Uint16
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required for the data being placed in this array.	1-10,000

#### Example

```
// Data Arrays
Data_Arrays
Data_Array_Name,
p1c1_devices,
p1c1_function,
p1c69_restart,
p1c69_function,
p1c90_reset,
p1c90_function,
```

	Data_Array_Format,	Data_Array_Length
p1c1_devices,	Bit,	1
p1c1_function,	Byte,	1000
p1c69_restart,	Bit,	1000
p1c69_function,	Byte,	100
p1c90_reset,	Bit,	100
p1c90_function,	Byte,	100

## 5.2. Server Side Connection Descriptors

Section Title	Function	Legal Values
Connections		
Column Title		
Port	Specify which port the device is connected to the FieldServer	P1-P8 <sup>2</sup>
Baud*	Specify baud rate	150, 300, 1200, 2400, 4800, <b>9600</b> (Vendor limitation)
Parity*	Specify parity	<b>None</b> (Vendor limitation)
Data_Bits*	Specify data bits	<b>8</b> (Vendor limitation)
Stop_Bits*	Specify stop bits	<b>1</b> (Vendor limitation)
Protocol	Specify protocol used	Secutron
Handshaking*	Specify hardware handshaking	None

### Example

// Server Side Connections				
Connections				
Port, P1,	Protocol, Secutron,	Baud, 9600,	Parity, None,	Handshaking, None

## 5.3. Server Side Node Descriptors

Section Title	Function	Legal Values
Nodes		
Column Title		
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	Generic Node ID unique to port. Also used as panel number.	0-9
Protocol	Specify protocol used	Secutron
Port	Port to which node belongs	P1-P8

### Example

// Server Side Nodes				
Nodes				
Node_Name, Node_1,	Node_ID, 1,	Protocol, Secutron,	Connection, P1	

<sup>2</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

## 5.4. Server Side Map Descriptors

### 5.4.1. FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of the Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in "Data_Array" section
Function	Function of Client Map Descriptor	Server
Node_Name	The name of the Node the Map Descriptor is attached to	One of the Node Names specified in "Server Node Descriptor" above
Length	The number of devices on the circuit the Map Descriptor must represent	0-999
Block_Type*	Used to enable packed bit when using a Packed_Bit Data Array on the Map Descriptor	Flag, -

### 5.4.2. Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Secutron_Type	Identifies the type of data the Map Descriptor points to	Device, Function
Secutron_Circuit	Identifies the circuit or panel zone	0-99

### 5.4.3. Map Descriptor Example.

<b>Map_Descriptor_Name,</b>	<b>Scan_Interval,</b>	<b>Data_Array_Name,</b>	<b>Data_Offset,</b>	<b>Function,</b>	<b>Node_Name,</b>	<b>Secutron_Type,</b>	<b>Secutron_Circuit,</b>	<b>Length</b>
P1c1_devices,	-	P1c1_devices	0,	Server,	Node_1,	device,	1,	200
P1c1_function,	-	P1c1_function	0,	Server,	Node_1,	function,	1,	200
P1c69_restart,	-	P1c69_restart,	0,	Server,	Node_1,	device,	69,	100
P1c69_function,	-	P1c69_function,	0,	Server,	Node_1,	function,	69,	100
P1c90_reset,	-	P1c90_reset,	0,	Server,	Node_1,	device,	90,	100
P1c90_function,	-	P1c90_function,	0,	Server,	Node_1,	function,	90,	100

A Data Array name must be one found under Data\_Arrays. Data from a circuit event will be stored in the array at Data\_Array\_Offset.

Any unique name. Name will appear in Bridge Map Descriptor status information screens.

Function may not be read or write since it implements a Server. Function may only be Server.

A name given to the device to which the FieldServer is connected.

The circuit or zone number used to trap incoming events into the Map Descriptor. See Driver notes.

The number of devices on the circuit this Descriptor can detect for incoming events.

The type of data the Map Descriptor points to. Device data contains alarm statuses. Function data contains condition codes. Refer to Appendix A.1 for more information.

The circuit or zone number used to trap incoming events into the Map Descriptor. See Driver notes.

## Appendix A. Advanced Topics

### Appendix A.1. Event Function or Condition Codes

When the Client driver receives an event from the panel, it stores the event's function or condition code in the Secutron\_Type=function Map Descriptor's Data Array at the device number's offset,

Example:

Device=3 on Circuit=1 goes into alarm – the Panel reports an event with function code A (Decimal 65) The Map Descriptor Secutron\_Type=device, Secutron\_Circuit=1 will set a bit and store the event function code A (Decimal 65) at offset 3 in the Map Descriptors' Data Array.

The Secutron\_Type=device Map Descriptors will set a bit if any of the events listed below occur on that circuit or zone. The bit will be cleared if the panel reports the event as cleared.

The full list of event function codes are:

ASCII value	Decimal	Meaning
A	65	Alarm
B	66	Bypass
C	67	Comlink
D	68	Duplicate Addressable Device
E	69	Maintenance Alert
G	71	Ground Fault
H	72	Hot Key Pressed
I	73	Illegal Addressable Device
J	74	Switch
M	77	Missing Addressable Device
N	78	Non-Fire/Non-Reporting
P	80	Pull Station Alarm
Q	81	Releaser Abort
R	82	Releaser Halt
S	83	Supervisory
T	84	Trouble or Parameter Change
U	85	Wrong Device Type
W	87	Waterflow Alarm

## Appendix A.2. Panel Circuit or Zone numbers

The panel groups events under circuit or zone numbers. When setting up a Client configuration it is necessary to create Map Descriptors to trap the events from the zones or circuits you want to see data from.

If an event is received that has no corresponding Map Descriptor defined, this will be reported on the RuiNet E screen. This is a useful configuration tool to ensure that all required data is stored.

Panel Zone	Description	Panel Zone	Description
0	General Alarm	53	Comlink 3
1	Initiating Circuit 1	54	Comlink 6
2	Initiating Circuit 2	55	Printer Port
3	Initiating Circuit 3	59	Addressable Modules Comlink
4	Initiating Circuit 4	60	Output (bell/releaser) Supervision
5	Initiating Circuit 5	61	Auxiliary Power Supervision
6	Initiating Circuit 6	64 <sup>1</sup>	Program Checksum
7	Initiating Circuit 7	65 <sup>1</sup>	Database Checksum
8	Initiating Circuit 8	66	AC Power
9	Initiating Circuit 9	67	Low Battery
10	Initiating Circuit 10	69 <sup>1</sup>	Program Restart
11	Initiating Circuit 11	71	Network Ground Fault
12	Initiating Circuit 12	80 <sup>2</sup>	Privilege Level
13	Initiating Circuit 13	81	1st Stage Alarm
14	Initiating Circuit 14	82	2nd Stage Alarm
15	Initiating Circuit 15	83	Switch
16	Initiating Circuit 16	84	Test Mode
17	Initiating Circuit 17	85 <sup>1</sup>	Erase Configuration
18	Initiating Circuit 18	86 <sup>3</sup>	Change Configuration
19	Initiating Circuit 19	87	Passcode Tamper
20	Initiating Circuit 20	88 <sup>1</sup>	Database Loaded
21	Initiating Circuit 21	89 <sup>1</sup>	Time/Date Change
22	Initiating Circuit 22	90 <sup>1</sup>	Hot Key Activation
23	Initiating Circuit 23	91 <sup>1</sup>	Memory Overflow
24	Initiating Circuit 24	92 <sup>1</sup>	Network Reboot
50	Network Verify	93 <sup>1</sup>	New Program
51	Comlink 1	94 <sup>1</sup>	Network Reboot Required who
52	Comlink 2	95 <sup>1</sup>	Network Reboot Required why

### Notes:

1. These zones do not restore.
2. Privilege Level 0 generates a restore signal; all others generate a trouble signal.
3. This zone does not restore and is not repeated sequentially in the archive.

### Appendix A.3. Panel Power-up, Restarts and Resets

When the panel powers up or is restarted it will send an event on zone/circuit 69 device 0.

When the panel is reset from the Hot Keys on the front panel, it will send an event on zone/circuit 90, device 9.

The Client driver will clear all the Secutron\_Type=device and Secutron\_Type=function Data Arrays when it traps any of these 2 events, provided the circuit 69 and 90 Map Descriptors have been set up as per the example Map Descriptors under the Client Side configuration instructions.

Other events for Circuits 69 and 90 are shown below for which a bit will be set in the Secutron\_Type=device Map Descriptor's Data Array at the device offset values shown below:

Circuit	Events	
Program Restart (Circuit 69)	000 – Power On/Hard reboot 001 - Watchdog 002 – Clock Monitor	003 – Illegal Instruction 004 – Unused Interrupt 005 – Orphan Interrupt
Hot Key Activation (Circuit 90)  Function number of the Hot Key pressed.	001 – Fire Drill 002 - Lamp Test 003 – Releaser Disconnect 004 – Relay disconnect 005 – Test mode 006 – Signal Disconnect 007 - Acknowledge 008 – Signal Silence 009 - Reset 010 – Common Disconnect 011 – General Alarm 012 - Halt 013 - Abort 013 – 2 <sup>nd</sup> Stage Inhibit 016 – Switch 1 On 017 – Switch 2 On 018 – Switch 3 On 019 – Switch 4 On 020 – Switch 5 On 021 – Switch 6 On 022 – Switch 7 On 023 – Switch 8 On 024 – Switch 9 On 025 – Switch 10 On	026 – Switch 11 On 027 – Switch 12 On 028 – Switch 13 On 029 – Switch 14 On 030 – Switch 15 On 031 – Switch16 On 032 – Switch 1 Off 033 – Switch 2 Off 034 – Switch 3 Off 035 – Switch 4 Off 036 – Switch 5 Off 037 – Switch 6 Off 038 – Switch 7 Off 039 – Switch 8 Off 040 – Switch 9 Off 041 – Switch 10 Off 042 – Switch 11 Off 043 – Switch 12 Off 044 – Switch 13 Off 045 – Switch 14 Off 046 – Switch 15 Off 047 – Switch16 Off 051 – Manual Restart

### Appendix A.4. Driver Messages, Error Messages and Optional Map Descriptors

The Client driver will report all events it receives from the panel on the RuiNet F screen.

If an event is received that has no corresponding Map Descriptor defined, this will be reported on the RuiNet E screen. This is a useful configuration tool to ensure that all required data is stored.